Amendment to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method comprising:

allocating a memory entry in a memory device included in a multithreaded engine to executable instructions stored in the multithreaded engine, the executable instructions to be executed on **[[**a**]]** the multithreaded engine included in a packet processor; **[[**and**]]**

including a unique identifier assigned to the executable instructions in a portion of the memory entry; and

maintaining a count of a number of threads included in the multithreaded engine that use the memory entry when the multithreaded engine executes the executable instructions.

2. (Cancelled).

3. (Original) The method of claim 1, further comprising:

maintaining a bit to represent availability of the memory entry for thread use.

4. (Currently Amended) The method of claim **[[**2**]]** 1 wherein maintaining the count includes incrementing the count to represent a thread initiating use of the memory entry.

5. (Currently Amended) The method of claim **[[**2**]]** 1 wherein maintaining the count includes decrementing the count to represent a thread halting use of the memory entry.

6. (Original) The method of claim 3 wherein maintaining the bit includes setting the bit to represent availability of the memory entry for thread use.

7. (Original) The method of claim 3 wherein maintaining the bit includes clearing the bit to represent unavailability of the memory entry for thread use.

8. (Original) The method of claim 3, further comprising:

checking the bit to determine the availability of the memory entry for thread use.

9. (Original) The method of claim 1 wherein the unique identifier includes four bits.

10. (Original) The method of claim 1 wherein the memory entry identifies a location in a local memory included in the multithreaded engine of the packet processor.

11. (Currently Amended) A computer program product, tangibly embodied in a machine-readable medium, the computer program product being operable to cause a machine to:

allocate a memory entry in a memory device included in a multithreaded engine to executable instructions stored in the multithreaded engine, the executable instructions to be executed on **[[a]]** the multithreaded engine included in a packet processor; **[[**and**]]**

include a unique identifier assigned to the executable instructions in a portion of the memory entry; and

maintain a count of a number of threads included in the multithreaded engine that use the memory entry when the multithreaded engine executes the executable instructions.

12. (Cancelled).

13. (Original) The computer program product of claim 11 being further operable to cause a machine to:

maintain a bit to represent availability of the memory entry for thread use.

14. (Currently Amended) The computer program product of claim **[[**12**]]** 11 wherein maintaining the count includes incrementing the count to represent a thread initiating use of the memory entry.

15. (Currently Amended) The computer program product of claim **[[**12**]]** 11 wherein maintaining the count includes decrementing the count to represent a thread halting use of the memory entry.

16. (Original) The computer program product of claim 13 wherein maintaining the bit includes setting the bit to represent availability of the memory entry for thread use.

17. (Original) The computer program product of claim 13 wherein maintaining the bit includes clearing the bit to represent unavailability of the memory entry for thread use.

18. (Original) The computer program product of claim 13 being further operable to cause a machine to:

check the bit to determine the availability of the memory entry for thread use.

19. (Original) The computer program product of claim 11 wherein the unique identifier includes four bits.

20. (Original) The computer program product of claim 11 wherein the memory entry identifies a location in a local memory included in the multithreaded engine of the packet processor.

21. (Currently Amended) A memory manager comprises:

a process to:

allocate a memory entry in a memory device included in a multithreaded engine to executable instructions stored in the multithreaded engine, the

executable instructions to be executed on **[[**a**]]** the multithreaded engine included in a packet processor; **[[**and**]]**

include a unique identifier assigned to the executable instructions in a portion of the memory entry;

maintain a count of a number of threads included in the multithreaded engine that use the memory entry when the multithreaded engine executes the executable instructions;

determine that the memory entry is no longer being used by a thread included in the multithreaded engine; and

decrement the count.

22. (Cancelled).

23. (Original) The memory manager of claim 21, further comprises:

a process to maintain a bit to represent availability of the memory entry for thread use.

24. (Currently Amended) A system comprising:

a packet processor to:

allocate a memory entry in a memory device included in a multithreaded engine to executable instructions stored in the multithreaded engine, the executable instructions to be executed on **[[**a**]]** the multithreaded engine included in a packet processor; **[[**and**]]**

include a unique identifier assigned to the executable instructions in a portion of the memory entry;

maintain a count of a number of threads included in the multithreaded engine that use the memory entry when the multithreaded engine executes the executable instructions;

determine the initiation of use of the memory entry by a thread included in the multithreaded; and

in response to the determining, increment the count.

25. (Cancelled).

26. (Previously presented) The system of claim 24 wherein the packet processor is further configured to:

maintain a bit to represent availability of the memory entry for thread use.

27. (Currently Amended) A network forwarding device comprising:

an input port for receiving packets;

an output for delivering the received packets; and

a network processor to:

allocate a memory entry in a memory device included in a multithreaded engine to executable instructions stored in the multithreaded engine, the executable instructions to be executed on **[[a]]** the multithreaded engine included in a packet processor; **[[and]]**

include a unique identifier assigned to the executable instructions in a portion of the memory entry; and

maintain a count of a number of threads included in the multithreaded engine that use the memory entry when the multithreaded engine executes the executable instructions.

28. (Cancelled).

29. (Previously Presented) The network forwarding device of claim 28, wherein the network processor is further configured to maintain a bit to represent availability of the memory entry for thread use.

30. (Currently Amended) A method comprising:

allocating a 32-bit long content-addressable-memory (CAM) entry to an executable microblock to be executed on a multithreaded microengine included in a

network processor, the 32-bit long CAM entry and the executable microblock located in the multithreaded engine; and

including a 4-bit long unique identifier assigned to the executable microblock in a portion of the CAM entry.

31. (Original) The method of claim 30, further comprising:

maintaining a count of threads included in the multithreaded microengine that use the CAM entry.

32. (Original) The method of claim 30, further comprising:

maintaining a bit in a status register to represent availability of the CAM entry to identify a local memory location.

33. (Previously Presented) The method of claim 1, wherein the memory entry comprises a content-addressable memory entry.

34. (Previously Presented) The computer program product of claim 11, wherein the memory entry comprises a content-addressable memory entry.

35. (Previously Presented) The memory manager of claim 21, wherein the memory entry comprises a content-addressable memory entry.

36. (Previously Presented) The system of claim 24, wherein the memory entry comprises a content-addressable memory entry.

37. (Previously Presented) The network forwarding device of claim 27, wherein the memory entry comprises a content-addressable memory entry.